УДК 004.896:007.52

Мезенцева А.Д.

РЕАЛИЗАЦИЯ АЛГОРИТМА ДВИЖЕНИЯ ДЛЯ НАВИГАЦИИ НАЗЕМНЫХ РОБОТОВ НА МЕСТНОСТИ

Аннотация. Работа посвящена реализации навигации наземного робота, используя по выбору пользователя один из алгоритмов движения: релейного, пропорционального или пропорционально-дифференциального регуляторов. Программная реализация представлена как узел ROS (Robot Operating System).

Ключевые слова: роботы, навигация, движение, регуляторы.

Abstract. The work deals with implementation of a ground robot navigation supporting user choice of an algorithm to use. Relay, proportional, and proportional-differential controllers are all supported. Programmatic implementation was devloped as a ROS (Robot Operating System) node.

Keywords: robots, navigation, movement, regulators.

Введение

Современный этап развития общества характеризуется широким внедрением информационных технологий. Роботы облегчают нашу жизнь и используются во многих сферах жизни. Органы чувств роботам заменяют датчики, однако они бесполезны без особого алгоритма, который позволит ориентироваться на местности и двигаться к заданной цели.

На базе Уральского федерального университета уже два года ведется проект «Полигон», направленный на реализацию групповой системы управления роботами. Для тестирования системы используются гусеничные роботы, которые могут получать команды из внешней системы управления. Для реализации бортовой системы управления (БСУ) требовалось разработать алгоритм, который позволил бы роботам двигаться в заданную точку.

Всего существует не так много типов регуляторов [1], которые применяют для управления движениями, всего их три: релейный, пропорциональный и пропорционально-дифференциальный регуляторы, есть и другие типы, но их применяют достаточно редко. Все регуляторы отличаются по своим характеристикам и сложности реализации.

Представленная работа посвящена решению практической задачи: реализовать движение наземного робота к цели, при этом по выбору пользователя использовать один из алгоритмов релейного, пропорционального или пропорционально-дифференциального регуляторов. Для реализации задачи использовалась операционная система Ubuntu с установленной надстройкой Robot Operating System (ROS).

Образцы и методика эксперимента

В проекте «Полигон» используется гусеничный робот (Рис. 1), который представляет собой конструкцию, состоящую из гусеничной платформы, аккумуляторной батареи, лазерного сканирующего дальномера RPLIDAR, радиомодуля, одноплатового компьютера Raspberry Pi и контроллера Pixhawk, реализующего функционал автопилота.

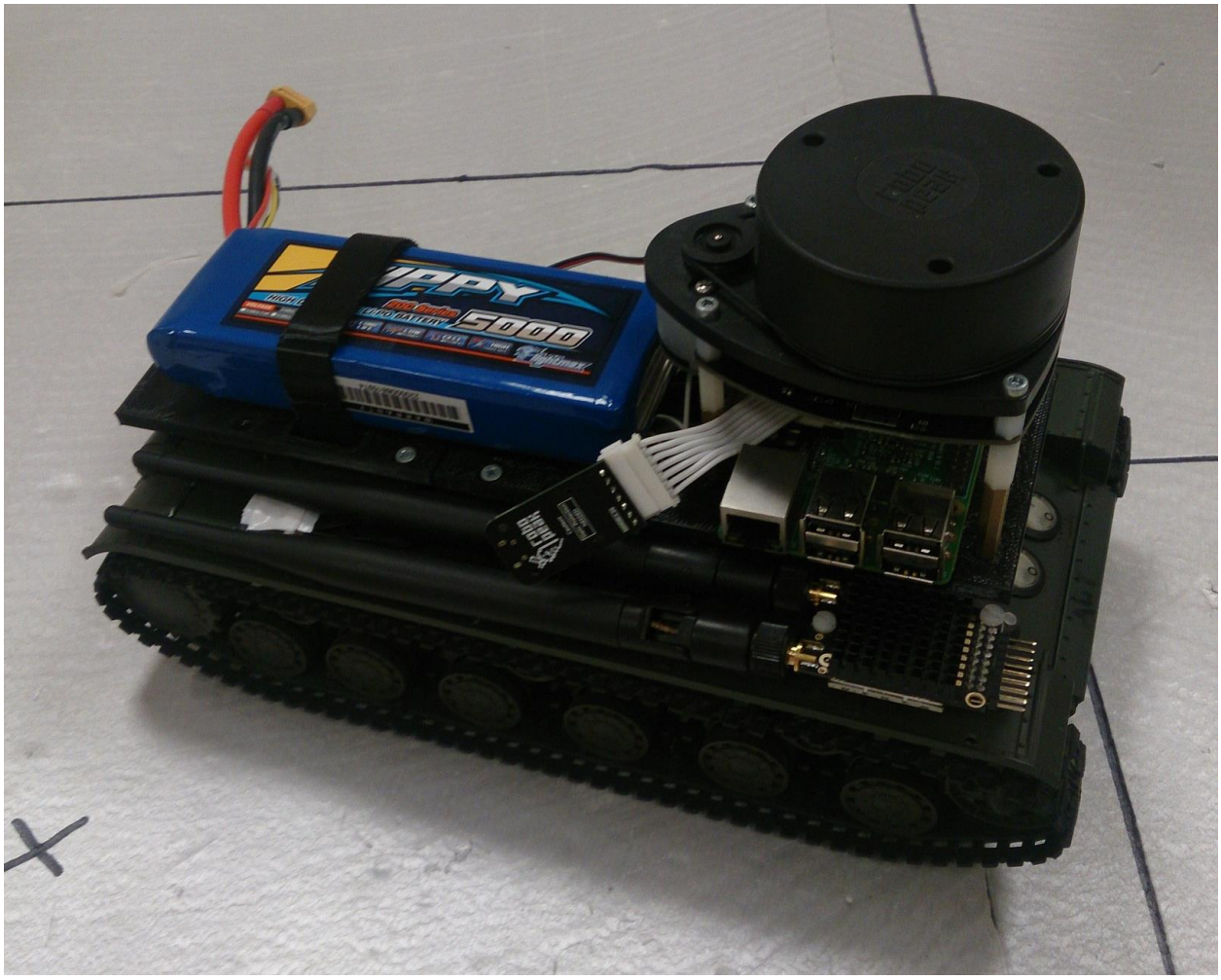

Рисунок 1 – Гусеничный робот

Лазерный сканирующий дальномер RPLIDAR необходим для построения карты помещения и позиционирования на ней робота посредством метода одновременной локализации и построения карты (SLAM) [2]. Дистанция работы дальномера шесть метров, результат работы представляет собой массив расстояний до препятствий.

Радиомодуль используется для передачи команд с внешнего управления на борт робота для дальнейшей их интерпретации при помощи БСУ, а также передачи обратно состояний робота. Для передачи используется протокол MAVLINK.

Компьютер Raspberry Pi используется для развертывания БСУ. На нем установлены операционная система Ubuntu и фреймворк ROS. Контроллер Pixhawk используется как вспомогательное устройство, позволяющее абстрагироваться от деталей электротехнической реализации робота.

БСУ реализована как программа на фреймворке ROS. Каждое приложение в ROS состоит из узлов. Узлы являются исполнительными процессами, которые могут взаимодействовать с другими узлами посредством сообщений.

Чтобы описать, какие узлы входят в приложение, используются файлы запуска – launch файлы, в которых указываются имена исполняемых узлов. Кроме того, в launch файле могут передаваться параметры, с которыми запускаются исполняемые файлы, что позволяет изменять некоторые переменные алгоритма, избегая повторного построения рабочей области – это сильно экономит время.

В ходе практического исследования были изучены и реализованы три регулятора, которые пользователь может выбрать по своему желанию перед запуском робота.

Релейный регулятор основан на следующем алгоритме: на основании координат текущего местоположения и координат заданной цели вычисляется угол, на котором находится робот и расстояние до цели, а затем, в зависимости от этих данных, принимается решение, что нужно сделать роботу:

1) Стоять, если робот добрался до цели
2) Двигаться прямо, если робот повернут в сторону цели
3) Поворачивать направо или налево, чтобы взять курс на цель.

Пропорциональный регулятор (П-регулятор) реализован на основе управляющего воздействия, которое следит за состоянием робота и вырабатывает для него управляющие сигналы в соответствии с заданным законом управления. В П-регуляторе закон управления выражается формулой:

$$u = error \cdot kp \qquad (1)$$

где *error* – это ошибка между углом заданной цели робота и его текущим положением, а *kp* – коэффициент П-регулятора, константное значение, подбираемое индивидуально для каждого робота. Алгоритм устроен таким образом, что зная текущее действие, которое должен совершить робот, чтобы

добраться до цели, угол и координаты робота и цели, вычисляет ошибку error, а затем вырабатывает управляющее воздействие. Это воздействие, которое заставит робота при отклонении в движении от цели, развернуться (выровняться) и достичь заданной точки, подается на моторы. Притом, чем больше робот отклонился от заданного курса, тем больше будет ошибка, и с тем большей скоростью он будет возвращаться на курс.

Пропорционально-дифференциальный регулятор (ПД – регулятор) имеет почти такой же вид, что и пропорциональный. Отличие заключается в том, что в формуле (1) появляется дополнительное слагаемое, включающее сведения об ошибке прошлого цикла управления и коэффициент Д-регулятора, что влияет на плавность движения робота:

$$u_i = error_i \cdot kp + \left(error_i - error_{i-1}\right) \cdot kd$$

В результате работы была реализована БСУ, которая имеет следующую конфигурацию узлов (Рис. 2): «Lidar» отправляет массив значений в узел «SLAM», где происходит построение карты помещения и позиционирования на ней робота, узел «Управление движением» и «MAVROS».

Алгоритмы регуляторов реализованы в составе узла «Управление движением», где получаются координаты текущего положения робота и координаты заданной цели, вычисляет необходимые действия, которые нужно сделать, чтобы добраться до цели и, в зависимости от выбранного регулятора, подает команды на гусеницы робота. «MAVROS», отправляет команды на борт робота.

Рисунок 2 – Конфигурация узлов ROS

Перед запуском робота пользователь указывает в файле запуска значение параметра regular:

1 – релейный регулятор,

2 – пропорциональный регулятор,

3 – пропорционально-дифференциальный регулятор.

После выбора регулятора, происходит считывание значения, которое присваивается переменной algorithm – она и отвечает за выбранный регулятор. Благодаря использованию параметра в файле запуска, нет необходимости собирать весь пакет полностью из-за внесенных изменений.

Результаты и обсуждение

Реализованные алгоритмы использовались на гусеничном роботе, в роли «глаз» которого используется лазерный сканирующий дальномер RPLIDAR, который необходим для построения карты помещения и позиционирования на неё робота посредством SLAM (метод одновременной локализации и построения карты). Всё это позволяет наземному роботу передвигаться по местности, зная координаты заданной цели и координаты своего местоположения.

В ходе испытаний разработанные алгоритмы тестировались на гусеничном роботе. Благодаря регуляторам движения, наземный робот успешно двигался в заданную точку, используя бортовую систему управления.

Заключение

В результате проделанной работы был разработан и реализован программный модуль ROS для бортовой системы управления роботом, который обеспечивает движение наземного робота к цели. По выбору пользователя можно использовать один из алгоритмов релейного, пропорционального или пропорционально-дифференциального регуляторов.

Разработанный узел является уникальным.

Библиографический список

1. Типы регуляторов и законы регулирования. Информационный портал по АСУ ТП [Электронный ресурс] // Автоматизация и управление в технических системах : blog. – Режим доступа: http://automation-system.ru/main/15-regulyator/type-of-control.html.
2. Локализация и составление карты с помощью DP-SLAM [Электронный ресурс] // Разработка роботов : сайт. – Режим доступа: http://robot-develop.org/archives/3954.
3. Архитектура ROS [Электронный ресурс] // Роботоша : блог о робототехнике, электронике и алгоритмах. – Режим доступа: http://robotosha.ru/robotics/ros-architecture.html.
4. Работа с фреймворком ROS [Электронный ресурс]. – Режим доступа: http://www.ros.org.